

## WWWとデータベースサービスとの連携方式の検証

元田 敏浩、徳丸 浩二

日本電信電話株式会社 ソフトウェア研究所  
東京都武蔵野市緑町 3-9-11

電話：0422-59-4069

E-mail: motoda@canary.sl.cac.ntt.jp

"Identification of  
Method for  
Coordinating  
WWW and Database  
Services"

あらまし

インターネット上のシステムとして近年注目されているものの1つにWWWがある。一般にWWWでは公開する情報をあらかじめ固定的に用意しておくが、リアルタイム情報のニーズも多く、その情報源の1つとしてリレーショナルデータベースとの連携が望まれている。我々はWWWクライアントからのデータベースアクセスの実現性と問題点について実験により検証した。その結果、単純な検索は問題無いがより高度な連携に際してはWWWに無い「セッション」の概念が必要である事が判明した。そしてその1つの対策としてWWWでセッションを実現する簡易セッション方式についても提案する。

キーワード WWW、データベース、インターネット、マルチメディア、検証

### An experimental verification of relational database access over WWW

Toshihiro Motoda, Koji Tokumaru

NTT Software Laboratories

Nippon Telegraph and Telephone corporation

3-9-11 Midori-cho Musashino-shi

Tokyo 180 Japan

Phone: 0422-59-4069

E-mail: motoda@canary.sl.cac.ntt.jp

#### Abstract

The WWW is a popular multimedia browsing system on the Internet today. Not only static information, many people want dynamic and real-time information from WWW servers. The relational databases, which used as a component of information systems, are suitable for source of such a dynamic information. We tried to browse information on a relational database from WWW clients, and found that some of the complicated database access applications needs a concept of "session", although the current WWW mechanism doesn't have one. We propose a simple session management mechanism on WWW systems for such applications.

key words WWW, database, internet, multimedia, verification

BEST AVAILABLE COPY

## 1. はじめに

インターネット上で近年急速に利用が増えているシステムの1つにWWW(World Wide Web)がある。WWWは世界中に分散する情報を連結し世界規模のハイパーテキスト空間を作成するというコンセプトから開発されてきたシステムである。

HTTP(Hyper Text Transfer Protocol)と呼ばれるプロトコルをベースにして、これまでインターネットで開発され、培われてきたFTP, NNTP, TELNET等の各種プロトコルをも統合し、それらのサーバにも透過的にアクセス可能である点で、WWWはこれまでの単一プロトコルのみを対象にしたクライアント・サーバシステムとは異なるユニークなシステムである。

一方、当初は学術用ネットワークとして構築された国内インターネットも商用インターネットプロバイダの出現により商用利用が可能となった。インターネットによる顧客への直接サービス等の商用利用を考える上で、操作が容易で、インストール数が急速に増加しているWWWクライアントの利用は有効であると考えられる。

一般にWWWによる情報提供では、各種の情報源から変換された文書情報を基にして画像・音声、さらには他の文書情報等をリンクさせて一連の情報を作成する。変化が少ない内容に関してはこの方法でも十分にサービス可能であり、それゆえ多くの情報はこの形態により提供されている。しかしWWWサーバのアクセス傾向の分析等から、固定的な情報へのアクセスに比較して、ニュースや株価情報などの変化のある情報、検索サービス等の能動的な情報へのニーズが大きいと言われている。一方、現状の企業の情報システムは多くの場合リレーショナルデータベース等何らかのデータベースを中心にして構築される事が多い。これらのデータベースには企業活動の生きたデータが格納されており、それを加工し提供する事ができれば情報提供効果が高いと考えられる。またWWWからフォーム入力によりネットワーク経由で直接受注が可能なシステム等で企業の情報システムへと直接データを投入すれば、低コストで正確な受注システムも構成可能である。

このように、データベースとWWWクライアン

トとを結んだ形態での、いわゆる連携型サービスが望まれている。我々はWWWクライアントからリレーショナルデータベースへのアクセスが可能である事を実験を通して検証し、問題点を洗い出した。本稿では実験結果を報告するとともに、問題点とその解決案について提案する。

## 2. 要素技術

### 2.1 WWWについて

WWWの構成および接続タイミングを図2.1に示す。WWWはインターネットに接続されたクライアントとサーバから構成され、基本的な動作は次の通りである。

ユーザがクライアントを操作して、必要な情報のURL(Uniform Resource Locator)と呼ばれる、インターネット上の資源に一元的に付与された名前を指定する。URLは、大きく3つの部分に分かれる。(1)プロトコル指定部と、(2)サーバ指定部、そして(3)サーバ上の資源指定部である。クライアントではDNS(Domain Name Server)機構を用いて、まずサーバ名称(ドメイン名)から、サーバのIPアドレスを得る。

そしてプロトコル部がhttp:である場合、サーバに対して直接TCP/IPで接続を行い、取得したい情報のURLや自クライアントの受諾可能なデータの形式等の情報をサーバに送る。サーバは受け取ったURLを解釈し、対応する情報のうち最も適したものをMIME形式にてクライアントに送出し、その後コネクションを切断する。このプロトコル(HTTP)の特徴として、次の点が挙げられる。

- ・仕様公開されたオープンな規格である。
- ・要求毎にTCP/IPでの接続・切断を行う。
- ・テキスト、画像、音声等多種多様な情報を一元的に取得可能である。
- ・一部のコネクションレスプロトコルをカプセル化し包含可能である。

また、URLではプロトコル部分の表現に文字列を用いているため、組み合わせ的には任意のプロトコルが表現可能である。これはWWWで扱うプロトコルに対する将来にわたる拡張性を意味している。



置する。ゲートウェイサーバは WWWサーバとして待機し、WWWクライアントからの要求を受け、その要求を基に DBクライアントとして DBサーバにアクセスし、得られた結果を再び WWWクライアントに返却する。

WWWサーバが持つ外部スクリプト起動インタフェース CGI (Common Gateway Interface) と、VGUIDEのカスタムアプリケーション(AP)を利用する事で WWWサーバ、WWWクライアント、DBサーバともに無改造で実現可能である。

ただし、中継によるオーバーヘッドは避けられないため、性能的な問題点が懸念される。

### 方式3 - WWWクライアントに DBクライアント機能を持たせる

方式1の逆で、WWWクライアントに DBクライアント機能を追加した専用WWWクライアントが DBサーバに直接接続する。

既存 DBサーバには一切手を加えず、URLのプロトコル表記の1つに加えるという点が特徴である。

しかし、一般に DBクライアントは採用しているプロトコルが専用かつ非公開の場合が多い。また本方式の場合、WWWクライアントに対する改造が必要であり、公衆からのアクセスという前提を考えた場合、このように改造した WWWクライアントを広めるのは非常に困難である。WWWクライアントがプロトコル処理部分をモジュールとして外部から自由に組み込める機能を持つようになればモジュール化し、プロトコル処理部のみとして配布する等の可能性も考えられるが、現時点では不可能である。

本実験では、その目的からして容易な実現が望まれるところであり、また今回は連携可能性の検証が主体であり性能面での評価の必然性はないとの点から、方式2 (ゲートウェイサーバ設置方式) を採用した。

なお、ゲートウェイサーバ上の WWWサーバには入手性の面等から CERN httpd を用いた。

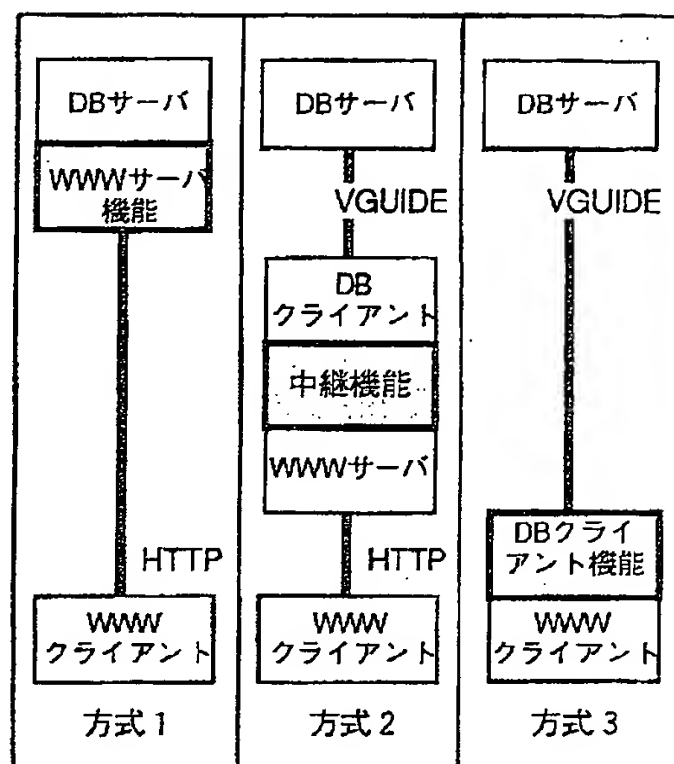


図3.1  
連携方式案

## 4. 実験

ここでは、WWWクライアントからのDBアクセスを行った実験についてその詳細を述べる。

### 4.1 実験1、実験2

WWW経由による基本的なデータベースアクセスおよび、テキスト形式での検索結果取得の確認を行った。

実験1では WWWクライアントからの要求でスクリプトおよびVGUIDE カスタムアプリケーション(AP)を起動し、VGUIDE カスタムAP内部に組み込んだ固定パターンの検索を行う。カスタムAPでは検索結果をスクリプトに渡し、スクリプトで検索結果をHTMLに変換して、WWWクライアントに送り、検索結果が表示される。

実験2は、SQL文による検索条件をWWWクライアントのフォーム入力により入力させ、渡された条件をカスタムAPへと渡し、以降実験1と同様に検索を行うことで指定された条件での検索結果を表示する。

### 4.2 実験3

グループ化および集約指定によるDB検索結果をテキストおよび円グラフとして表示する実験を

行った。

操作としては、(1) テーブル選択、(2) カラム選択、(3) データベース検索の3段階とした。

テーブル選択は、データベース中の全テーブルのうち検索に使用するテーブルを選択するもので、VGUIDEカスタムAPによりテーブル一覧を出力しWWWクライアントのフォーム入力で1つを選択させる。

カラム選択は、上記で選択されたテーブルに対して、別のVGUIDEカスタムAPにより当該テーブル中の全カラムを出力させ、同様にWWWクライアントのフォーム入力により1つ選択させる。

データベース検索では、上記選択されたテーブルおよびカラムよりSQL文を生成し、VGUIDEカスタムAPによりデータベースにアクセスする。カスタムAPより出力された検索結果は、実験1と同様にHTMLに変換されると同時に、ghostscriptを利用したグラフ化プログラムに渡してグラフィメーを生成し、先のHTMLに変換した検索結果に埋め込んでWWWクライアントに送り両者を合成した表示が得られた。

しかし、HTMLにおけるイメージの埋め込みは、イメージの実体は別の場所に存在し、HTML本体にはそのイメージのURLのみを埋め込み、WWWクライアントが両者を合成する事で実現されている。通常の、静的な組み込みの場合には固

定のURLを埋め込むため問題はないが、検索データにより異なるイメージが生成される場合に、は固定のURLを用いることができない。

そこで、検索を実行した際の時刻および実行したスクリプトのプロセスIDから一意の数字列を生成し一時ファイル名とし、当該名でグラフィメーを生成すると同時に当該名からURLを生成しHTMLに埋め込む事で解決した。

#### 4. 3 実験4

1つのDB検索結果の全体を表示する一覧形式および、個々のデータについて詳細に表示する個別形式の2種類のビューでの表示を試みた。

検索は固定パターンを用いた。検索対象は社員情報テーブルで、全件検索を指定した。

WWWクライアントから検索を指定されると、VGUIDEカスタムAPが起動され、検索が行われる。検索結果はスクリプトへと渡され、検索結果のうち「氏名」情報から一覧表示用のHTMLを生成し、同時に一覧表示の各項目には個別表示のページへのURLを付加する。また、検索結果からは検索結果の全データを使用して個別表示用のHTMLも同時に生成した。

個別形式のHTMLは実験3の場合と同様に時刻・プロセスID・レコード番号から一時ファイル名を作成し、その中に格納した。

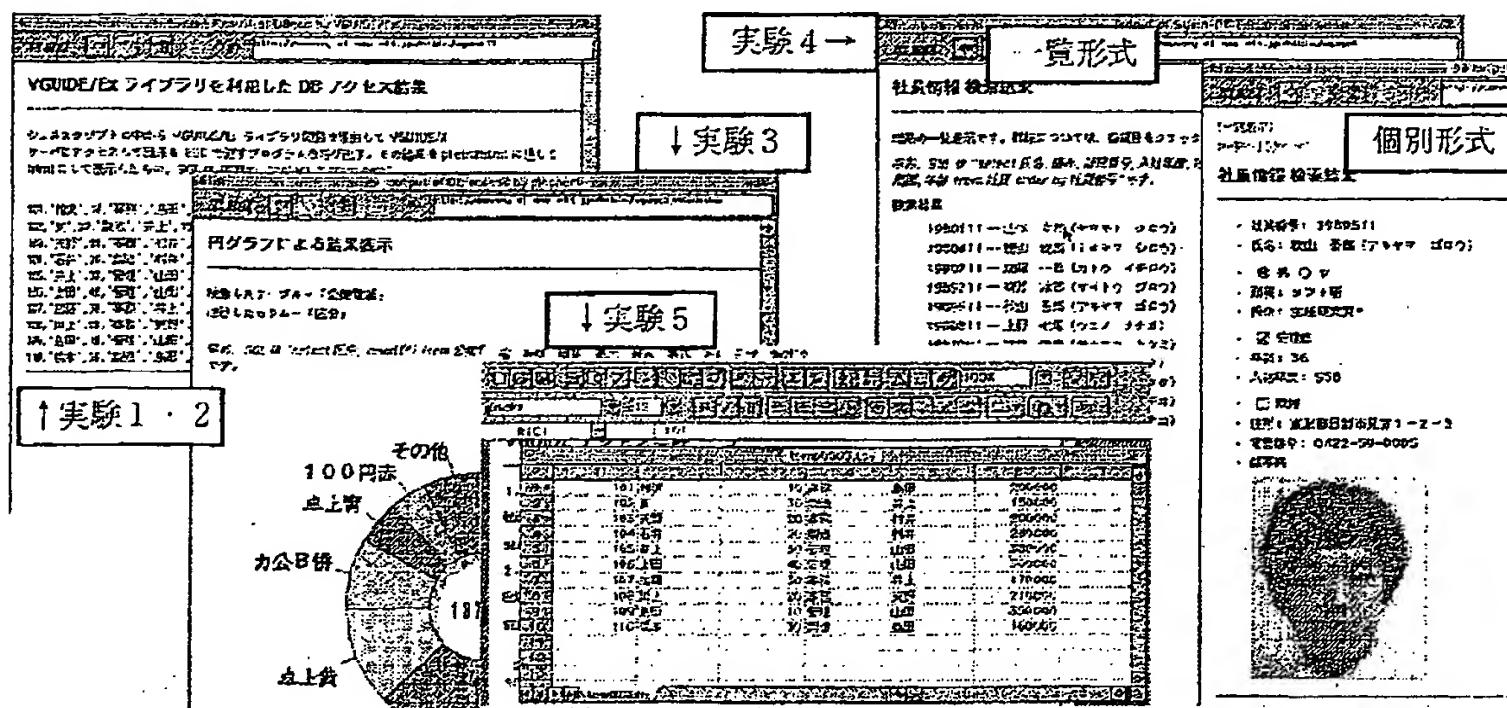


図4. 1 実験画面

は別途固定で用意し、それを埋め込む HTML を生成した。

#### 4. 4 実験5

DB 検索結果を WWW クライアントを経由して他のアプリケーションへと受け渡す動作の確認を行った。

HTTP では WWW クライアントに戻す電文のヘッダ部分には内容に関する情報が含まれている。CGI スクリプトでは、このうち「情報の種類」を示す「Content-Type:」ヘッダが制御可能であり、これを利用した。まず WWW クライアントから検索条件を指定して、検索を指示する。検索は、VGUIDE カスタム AP によって実行され、指定された出力形式を示すヘッダを出力した後、当該形式にて検索結果を出力する。例えば各カラム間をタブで区切ったデータの場合ヘッダは「Content-Type: text/tab-separated-values」である。

一方、WWW クライアントでは情報の種類に応じて外部アプリケーションを指定しておく。これにより、検索を指示するだけで外部アプリケーションに DB 検索結果を渡すことができる。

#### 5. 問題点

WWW を企業の情報公開や各種情報収集、さらには有料サービスや受注等に利用しようとした際の問題点について、これまでの実験を通じて明らかになった点について述べる。

##### 5. 1 セッション

データベースとのインタフェースを実現するに当たって最も問題となったのが、HTTP がアクセスの度に接続・切断を繰り返すいわゆる「コネクションレス」であるという点である。通常データベースでは、トランザクション状態、検索結果集合、ロック状態など DBMS が各種状態を保持しつつ処理が進行する。データベースを扱う場合サーバ・クライアントともこの状態を把握しつつアクセスを行う必要がある。

WWW から間接的にデータベースへのアクセスが行われる場合、データベースを直接扱う訳では

ないのでこれら全ての状態を保持する必要は無いと思われるが、個別の利用者毎に個別の検索結果等が必要な事から考えても何らかのセッション機能が必要であるものと予想される。

##### 5. 2 開発環境

今回の実験は、接続性の確認のための極めて小規模なものであったため、全てシェルスクリプトとCプログラムで個別作成した。しかし、実際にサービスが可能な規模のページを作成する場合には、何らかの開発の支援が必要になるものと思われる。

##### 5. 3 その他

その他、実際に商用サービスを行おうとした場合に問題となる問題点には以下のものが挙げられる。

##### ・WWW クライアントのキャッシュの問題

WWW クライアントでは通常一度アクセスしたデータをローカルにキャッシュしておくように実装してある。しかしながら、キャッシュ機能は必ず実装されているという保証は無くまた、キャッシュサイズの規定も無い。そのためキャッシュを前提に作成してしまうと、思わぬタイミングで再検索が実行された場合、以前の検索結果と矛盾したデータが検索されるなど、予期せぬ動作が起きる可能性がある。

これについては、例えば検索実行の際にもセッション機能を利用する等の対策が必要であると思われる。

##### ・セキュリティ

HTTP 1.0 では、データを全く暗号化せず伝送する。単なる不特定多数向けの情報公開ではなんら問題は無いが、会員制サービス提供・商品受注など何らかの識別子を用いた特定者への情報・サービス提供をも行おうとした場合、暗号化されていないために第三者による傍受や、それによる不正利用が行われる可能性がある。

しかし、この問題に対しては一部で既に対応に向かっており、公開鍵暗号を用いた伝送等が考慮

杉田様

杉田君に[ ]箇所のみ 翻訳をお願い。いま  
(p52の5.1, p53~54の6.1, 6.2の-部)



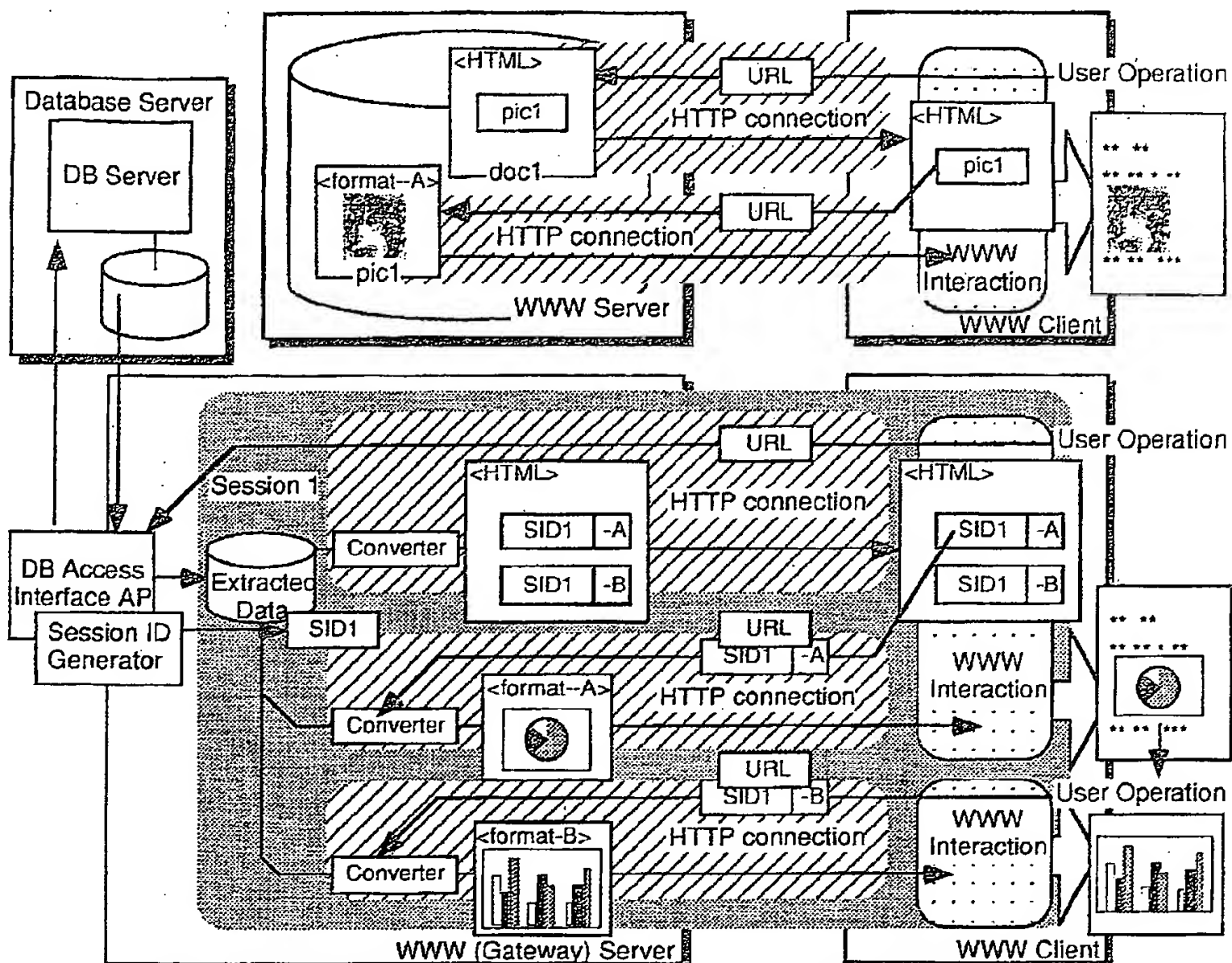


図6. 1 従来のHTTPアクセス(上)と簡易セッション方式によるDBアクセス(下)

されている。近い将来暗号化機能を内蔵したWWWサーバ・クライアントによる安全な通信が実現できるものと思われる。

#### ・パフォーマンス

商用サーバとして運用する場合、既存サーバ等の例から見てかなりのアクセス量があるものと想定される。現在のインプリメントでは、毎回スクリプト起動やプロセス起動およびコネクションを繰り返しているためかなりのオーバーヘッドが存在する。商用サーバとしての運用に供するにはより大量の処理をこなせるような仕組みが必要であると思われる。

#### 6. セッションについて

前述の問題点の通りHTTPではセッションの概

念を持たない。今回の実験ではこの点に関しての対策を一部実施したが、それについて検討する。

#### 6. 1 簡易セッション方式

実験3および4を実施する際に同一検索結果を異なるコネクション間で形式を変換して表示する必要がある。そのため前述の通り検索時刻および検索スクリプトのプロセスIDから一意な番号を生成し、それをいわゆるセッション識別子として、同一の検索結果にアクセスするための異なるコネクションを、識別するために使用した。ここではこの方式を「簡易セッション方式」と呼ぶことにする。

#### 6. 2 簡易セッション方式の問題点

上記簡易セッション方式では、実験レベルとし

てセッションを実現する事ができるが、実用サービスを目指す場合次のような点が問題になる。

#### ・第三者によるセッションの盗用

セッション識別子に時刻やプロセスID等規則性のあるものを用いている場合、第三者による推定が可能となる危険性がある。つまり、第三者による特定セッションの盗用が可能である。

これを防止するには乱数等を用いた特定困難なセッション識別子を用いる事が考えられる。しかし、確率はかなり低くなるが偶然性攻撃による盗用の危険性は残る。従って、実用的な盗用防止策としては、前述の今後充実されるであろうセキュリティ機能を併用し、利用者を限定した上で「利用者識別子」とこれまでの「セッション識別子」の組を、サーバの「内部セッション識別子」として保持する事で第三者によるセッション盗用を防止する事ができると考えられる。

#### ・セッション終了タイミングの特定

本方式ではセッション識別子がWWWサーバより払い出されて、WWWクライアントがそれをリンク情報の一部として保持し、WWWクライアント利用者が操作する事で、同一セッションにおいて別のデータが参照可能となる。しかし、伝送プロトコルにHTTPを用いている以上、一旦払い出されたセッション識別子がWWWクライアントでどのように保持されているか、あるいは廃棄されたのかといった情報はつかめない。そのため、サーバ側のセッションに関連した情報の保持期間つまり、セッション終了ポイントが不明確である。

現在のところ、有効な対策が無いため実験では一定時間でサーバ側のセッション情報を破棄している。すなわち、セッション開始後一定時間後にセッションを消滅させる事で回避している。

#### 7. おわりに

ゲートウェイサーバ設置によるWWWとデータベースの連携で、WWWクライアントからデータベースサーバにアクセス可能である事が実証された。これらの新たな形態による情報提供でより多

様な情報提供が可能になると思われる。

本実験はWWWに関するものであり、容易に外部からアクセス可能なものである。今後はここで実験した形態での公開実験を行い、実用性の検証を行うと同時に更新系も含めたデータベースアクセスとWWWとの連携についても検証をすすめた

#### 8. 参考文献

- [1] "The World Wide Web", The World Wide Web consortium, URL=<http://www.w3.org/hypertext/WWW>
- [2] "Hypertext Transfer Protocol"(HTTP), The World Wide Web consortium, URL=<http://www.w3.org/hypertext/WWW/Protocols/Overview.html>
- [3] "WWW Names and Addresses, URIs,URLs, URNs", CERN, URL=<http://info.cern.ch/hypertext/WWW/Addressing/Addressing.html>
- [4] "Hyper Text Markup Language (HTML)", CERN., URL=<http://info.cern.ch/hypertext/WWW/MarkUp/MarkUp.html>
- [5] "The Common Gateway Interface", NCSA, URL=<http://hoohoo.ncsa.uiuc.edu/cgi/overview.html>
- [6] "W3C Security Resource Page", URL=<http://www.w3.org/hypertext/WWW/Security>
- [7] "RSA Data Security, Inc Home Page", URL=<http://www.rsa.com/>
- [8] UniSQLホームページ、オンライン博物館。NTTデータ通信(株), URL=<http://unisql.www.nttdata.jp>
- [9] "WWW Access to Relational Database", URL=<http://info.cern.ch/hypertext/WWW/RDBGate>
- [10] "CERN httpd 3.0 Guide", CERN., URL=[http://www.w3.org/hypertext/WWW/Daemon/User\\_3.0](http://www.w3.org/hypertext/WWW/Daemon/User_3.0)
- [11] NTTソフトウェア株式会社:「VGUIDE/Eライブラリリファレンス」第3版



# An Experimental Verification of Relational Database Access Over WWW

Toshihiro Motoda and Koji Tokumaru  
NTT Software Laboratories  
Nipponn Telegraph and Telephone Corporation  
3-9-11 Midori-cho, Musashino-shi,  
Tokyo 180, Japan  
Phone: 0422-59-4069  
E-mail: motoda@canary.sl.cac.ntt.jp

## Abstract:

The WWW is a popular multimedia browsing system on the Internet today. Not only static information, many people want dynamic and real-time information from WWW services. The relational databases, which used as a component of information systems, are suitable for source of such a dynamic information. We tried to browse information on a relational database from WWW clients, and found that some of the complicated database access applications needs a concept of "session", although the current WWW mechanism doesn't have one. We propose a simple session management mechanism on WWW systems for such applications.

Keywords: WWW, database, internet, multimedia, verification

## 2. Elemental Technology

### 2.1 WWW

The structure of the WWW and its connectivity are shown in Fig. 2.1. The WWW is made up of clients and servers both connected through the Internet and functions basically as follows.

When a user manipulates a client, the user have to specify a particular name, which is given uniquely to the Internet resource and which is called URL (Uniform Resource Locator) for required information. The URL is broadly classified into three units: (1) Protocol Locator, (2) Server Locator, and (3) Server Resource Locator. The client makes use of the Domain Name Service mechanism to acquire the IP address of the server from the server's denomination (domain name).

If the protocol locator is http:, the client makes a direct connection with the server according to the TCP/IP and provides the server with the URL of the desired information and information on the data format acceptable to the client. The server then interprets the incoming URL and then send the most relevant information to the client in form of the MIME messages, followed by disconnection. This protocol (HTTP) has the following features.

- Publicly open standard, the specification of which has been published.

- Connection and disconnection according to the TCP/IP is carried out each time a request is made.
- A variety of information including texts, images and voices can be acquired monogenically.
- A certain connectionless protocol can be included having been capsulated.

Also, since the URL has a letter string to express a particular protocol, any arbitrary protocol can be expressed in terms of combination. This allows the URL to be expanded in the future to cope with the protocol used the WWW.

## 2.2 VGUIDE

The VGUIDE (Visual and General User Interface for Database Environment) was chosen as a database function for connection with the WWW. This VGUIDE was developed by NTT and is a middleware for the development of database-oriented custom application programs and systems based thereon. It provides the client-server type database access functionalities under the multi-vender environment and a development platform for the custom application utilizing these functionalities. The structure of this VGUIDE is schematically shown in Fig. 2.2. The client and the server can accommodate the database and the custom application, respectively and it makes possible to develop a database access system in an open manner from a side tool for individual businesses to a nationwide system.

## 3. Coordination System

Where reference is made from the WWW client to the data stored in the database (DB), since the both are a client-server type, the following coordination of the combinations can be contemplated in view of the respective structures of the DB and WWW systems.

System 1: The DB server concurrently serves as the WWW server.

The DB server is added with a mechanism for interpreting the HTTP to enable it to function as a WWW server.

As compared with any other systems, overhead hierarchies are so small that a high performance can be obtained.

However, since the information model differs between the WWW and the DB server, not only is it necessary for the flow of information between those systems to be properly switched, but also the DB server must be remodeled and, thus, more work is required than to newly construct the WWW server.

System 2: Installation of the Gateway Server

The gateway server for converting the protocol between the WWW client and the DB server is installed. The gateway server is used to stand ready for the WWW server, receive a request from the WWW client, make access to the DB server as a DB client based on the request and return the access result back to the WWW client.

When the external script activated interface CGI (Common Gateway Interface) possessed by the WWW and the custom application (AP) of the VGUIDE are utilized, the WWW server, WWW client and DB server can be realized with no need to be remodeled.

However, since the overhead resulting from the relay is unavoidable, problems on the performance appear to arise.

System 3: To Allocate the Function of the DB Client to the WWW client

This is reverse to the System 1 and the dedicated WWW client to which the DB client functionality to the WWW client is connected direct with the DB server.

This is featured by an addition to the protocol description of the URL without the existing DB server being modified.

However, it is often that the protocol generally employed in the DB client is special and not open to the public. In the case of this system, modification is necessary to the WWW client and, considering the necessity of making access from the public, it is extremely difficult to circulate the modified WWW client. If the WWW comes to have a function of freely incorporating the processing portion of the protocol externally into a module, it can be contemplated that it be so incorporated into a module so that only the protocol processing portion can be circulated, but it is impossible at this time.

In this experiments, the system 2 (Installation of the Gateway Server) was employed because since it can readily be realized, the purpose of the experiments is to verify the possibility of coordination and no evaluation is needed with respect to the performance.

In view of the availability, the CERN httpd was employed in the WWW server on the gateway server.

#### 4. Experiments:

The details of the experiments conducted to make access from the WWW client to the DB will be described.

##### 4.1 Experiment 1 and Experiment 2

Verification was made of the basic access to the database through the WWW

and results of search in the text format.

In this Experiment 1, the script and the VGUIDE custom application (AP) were started up in response to a request from the WWW client and a fixed pattern incorporated in the VGUIDE custom AP was searched. Through the custom AP, the search result was delivered to the script, and the search result in the script was converted into the HTML language, which was in turn transmitted to the WWW client with the search result displayed.

In the Experiment 2, search conditions in the SQL language were inputted in response to the form input of the WWW client and were then delivered to the custom AP, followed by the search made in a manner similar to that in the Experiment 1 to display the search conditions under a specific condition.

#### 4.2 Experiment 3

The experiment was conducted to display the DB search results, based on grouped and integrated assignment, in the text and the circle graph.

For manipulation, three stages were taken; (1) selection of the table, (2) selection of the column and (3) database search.

The table selection is to select one of the tables in the database, that is used for the search and is accomplished by outputting the look-up table from the VGUIDE custom AP so that the form input of the WWW client can select the requisite table.

The column selection is to causing the different VGUIDE custom AP to output all of the columns in the table selected in the manner discussed above, so that the form input of the WWW client can select one of the columns.

During the database search, the SQL is generated using the selected table and the selected column and the VGUIDE custom AP makes access to the database. The search result outputted by the custom AP is not only converted into the HTML in a manner similar to that in the Experiment 1, but delivered to the graph generating program utilizing the ghostscript to generate a graphic image, which is in turn embedded in the search result in the previously converted HTML format and then forwarded to the WWW client to provide a display combining the both.

However, embedment of the image in the HTML is accomplished by, while the image itself exist in a different site, only URL of such image is embedded in the HTML itself so that the WWW client can combine the both. While in the static embedment no problem occurs since the fixed URL is embedded, no fixed URL can be used where different images are generated from the search data.

In view of this, a unique string of numbers is generated using, and stored as a temporary file name, the time of execution of the search and the process ID of the executed script and the image file and the URL are generated under such file name and

are then embedded in the HTML.

#### 4.3 Experiment 4

An attempt is made to achieve a display under two kinds of views in different formats for displaying the details of the individual data and in a look-up format for displaying the single DB search result in its entirety.

For the search, the fixed pattern was employed. An object to be searched was an employee information table and the entire search was specified..

When the search is specified from the WWW client, the VGIDE custom AP was started to conduct the search. The search result was delivered onto the script and, out from the search results, the look-up HTML was generated using the "name of the employee" information and, at the same time, the URL for the individual display was added to each item of the look-up display. Also, from the search results, the HTML for the individual display was generated using all data of the search results.

The HTML for the individual display was generated by generating a temporary file name from the time·process ID·record number in a manner similar to that in the Experiment 3 and was stored therein.

#### 4.4 Experiment 5

Verification was conducted to ascertain the operation to deliver the DB search results to the other application through the WWW client.

The HTTP contains information concerning the contents in a header of the telegraphic message to be returned to the WWW client. With the CGI script, the "Content-Type" indicative of the "type of information" is controllable and this was utilized. In the first place, search conditions were specified from the WWW client to assign the search. The search was carried out by the VGUIDE custom AP and, after the header descriptive of the designated output type has been outputted, the search results are outputted in that type. For example, in the case of the data, in which a tab is used to separate the columns, the header is "Content-Type: text/tab-separated-values".

On the other hand, in the WWW client, an external application is specified depending on the type of information. In this way, mere assignment of the search is effective to deliver the DB search results onto the external application.

### 5. Problems

With respect to the problems which occur when the WWW is utilized for publication and collection of the various information on the industry and also for pay services and acceptance of an order, we will describes what has been clarified through those experiments.

### 5.1 Session

The legitimate problem associated with realization of the interface with the database lies in a so-called “connectionless”, in which HTTP connection and disconnection are repeatedly made during the access. In the ordinary database, DBMS (database management system) retains various conditions such as transaction state, set of search results and locked state while performing the process. In the case of the database being handled, it is necessary to make an access while the server and the client recognize those conditions.

When access is indirectly made from WWW to the database, the database is not handled directly and, accordingly, it appears that all of those conditions need not be retained, but some kind of session function or another appears to be necessary as the individual search results or the like are necessary for each individual user.

### 5.2 Development Environment

The experiments we had now conducted were on a small scale and, therefore, all were individually prepared with shell scripts and C-programs. However, where in practice pages are to be generated, which accommodate the services, it appears some development assistances are required.

### 5.3 Others

The following enumerates some problems which appear to arise when the services on a commercial scale are carried out.

- Problem on cache of the WWW client

The WWW client is provided with a cache for locally storing data once accessed. However, no warrant is available that the caching function is necessarily mounted and no cache size is specified. For this reason, if something is generated on the assumption that the cache is employed, there is a high possibility that an unexpected operation may occur such as display of data ambiguous with the previous search result when the search is again conducted at an unexpected timing.

To alleviate this problem, it appears that some countermeasures such as utilization of the session function are required even when the search is conducted.

- Security

With the HTTP 1.0, the data are transmitted without being encrypted. Although this is not problematic so far as the information is published to the unspecified public in general, unauthorized utilization of the unencrypted data such as tapping by the third party will occur particularly where supply of services and information to specific persons utilizing any identifier such as, for example, the membership services and the ordering of goods are concerned.



However, this problem appears to be partly remedied and the public key encryption system is currently considered. In the future, it appears that a secure WWW server-client communication will take place using the encryption system.

- Performance

Where the server is operated as a commercial server, it is suspected that the case with the existing servers suggests a substantial amount of accesses will take place. The current implement requires a substantial amount of overheads since the script start-up and the process start-up and connection are repeated for each time. Any framework appears to be necessary so that a substantial amount of processing can be handled where it is provided for the commercial server.

## 6. Session

As the foregoing problems indicate, the HTTP has no concept of “session”. During this experiment, portion of countermeasures with respect to this problem has been carried out, which will now be discussed.

### 6.1 Simple Session System

When Experiments 3 and 4 are conducted, it is necessary to display the same search results after they have been converted in format between different connections. For this reason, the unique number must be generated based on the search time and the process ID of the search script as discussed previously, and is then used as a so-called session identifier to identify the different connections for access to the same search results. This system is herein referred to as the “Simple Session System”.

### 6.2 Problems in Simple Session System

In the simple session system discussed above, the session can be realized on an experiment level, but the following problems will be highlighted where the simple session system is employed on practical services.

- Appropriation by the Third Party

In the case where something regular such as time and process ID is used for the session identifier, there is the risk that it may be inferred by the third party. In other words, the third party may appropriate the particular session.

To alleviate this problem, it may be suspected to use the hard-to-identify session identifier comprised of random numbers. However, although the probability may be low, there remains a problem associated with the appropriation caused by the chance attack. Accordingly, for practical countermeasures against the appropriation, it may be suspected that the appropriation of the session by the third party may be achieved by the utilization of the security functions, which would be sophisticated in the future as discussed previously and, at the same time, by retaining a set of the “user

identifier” for limited users and the conventional “session identifier” as a “internal session identifier” of the server.

- Determination of Session Terminating Timing

In this system, the session identifier is generated by the WWW server and the WWW client retains it as link information so that when the WWW client user utilize it, reference can be made to different data in the same session. However, since the HTTP is used as a transmission protocol, no information is available how the session identifier once deacquired is retained in the WWW client or how the session identifier is disposed of. For this reason, the period during which the information associated with the session on the part of the server is retained, that is, the point of termination of the session is not clear.

At present, since no effective countermeasure is available, in this experiment the session information on the part of the server is discarded after a predetermined length of time. In other words, the session is extinguished after a predetermined length of time subsequent to the start of the session.

## 7 Conclusion

It has been confirmed that coordination between the WWW and the database makes it possible for the WWW client to make access to the database server due to the installation of the gateway server. It appears that a variety of information can be provided for as information is provided according to those new systems.

The experiment conducted is associated with the WWW and access can be made easily from the outside. We wish that in the future, public experiments on the system experimented can be made to verify coordination between the WWW and the database access, including verification of the practical applicability and those updated.

**This Page is Inserted by IFW Indexing and Scanning  
Operations and is not part of the Official Record**

**BEST AVAILABLE IMAGES**

Defective images within this document are accurate representations of the original documents submitted by the applicant.

Defects in the images include but are not limited to the items checked:

- ☐ **BLACK BORDERS**
- ☐ **IMAGE CUT OFF AT TOP, BOTTOM OR SIDES**
- ☐ **FADED TEXT OR DRAWING**
- ☐ **BLURRED OR ILLEGIBLE TEXT OR DRAWING**
- ☐ **SKEWED/SLANTED IMAGES**
- ☐ **COLOR OR BLACK AND WHITE PHOTOGRAPHS**
- ☐ **GRAY SCALE DOCUMENTS**
- ☐ **LINES OR MARKS ON ORIGINAL DOCUMENT**
- ☐ **REFERENCE(S) OR EXHIBIT(S) SUBMITTED ARE POOR QUALITY**
- ☐ **OTHER:** \_\_\_\_\_

**IMAGES ARE BEST AVAILABLE COPY.**

**As rescanning these documents will not correct the image problems checked, please do not report these problems to the IFW Image Problem Mailbox.**